RDFox v7.1a Upgrade Instructions

The RDFox v7.1a patch update includes a fix for a dictionary compaction bug that could cause the software to derive incorrect facts (RD-1794), a fix for a crash that occurred when an RDFox server with one or more offline data stores was shut down explicitly using a Java or C API call (RD-1785), and a new feature that allows the parameters for password hashing to be explicitly specified during server initialization to bypass RDFox’s routine for automatically determining them. The release notes for the release are available here.

All users are advised to upgrade to RDFox v7.1a as soon as possible. Users who:
- will repopulate their v7.1a server from primary source data,
- have never imported Datalog rules into any data store in their server,
- are currently running a version of RDFox lower than v6.3, or
- do not use persistence

can safely upgrade to the new version directly, paying attention to guidance in the relevant migration guides (6.0, 7.0, and 7.1). All others users are advised to first read the description of the dictionary compaction bug in section Error! Reference source not found. of this document and to follow the instructions in section 2 to ensure that their RDFox instances are free from any possible effect of this bug.

If any of the information in this guide is not clear, or if you require additional support with any aspect, please reach out to Oxford Semantic Technologies support via your usual channel.

1. Detailed description of the dictionary compaction bug

The dictionary compaction bug is present in RDFox versions between 6.3 and 7.1 (inclusive), included lettered releases in that range.

The bug may derive incorrect facts after the following sequence of events occurs in a data store:

1) A rule is imported that:
   - contains one or more resources (IRIs or literals) that has not previously been introduced into the data store, and
   - does not match any data in the store.
2) The data store is compacted.
3) A read/write transaction is started and either rolled back or committed.

In versions of RDFox affected by the bug, the rule imported in step 1 now contains references to either the wrong resources or to non-existent resources. If the incorrect resource references appear in the head of the rule and the rule does eventually match data in the store, it will derive the wrong facts. If the incorrect references appear in the body of the rule, the rule may potentially match data that it should not, resulting in incorrect facts.

One side effect of the bug is that attempting to explain the incorrect facts in the RDFox console either crashes the UI or gives an explanation that does not follow.
2. Recovering from the dictionary compaction bug

RDFox v7.1a can load server directory content persisted by RDFox versions 7.0 and 7.1 directly, however restarting the server using version 7.1a will not automatically correct any incorrectly derived facts that have been persisted previously so additional actions are needed to ensure that the results of reasoning are correct. In all cases, transcribing the content of a server and importing it into a fresh v7.1a server will result in a fully correct materialization, but in some cases this step can be avoided when upgrading from v7.0 or v7.1. The following flow chart can be used to guide you through the upgrade to version 7.1a in a way that guarantees it will be free from any incorrect facts introduced by the bug.

Notes from flow chart

A: To transcribe a server, use the `transcribe` shell command.

B: The table scanning query is `SELECT * { { ?S ?P ?O } UNION { GRAPH ?G { ?S ?P ?O } } }`. The runtime for this query will be proportionate to the number of triples and quads in the data store. For large stores, it is recommended to run this query in the RDFox shell with the `output shell variable` set to `null` which will ensure that the query evaluation is not slowed down by printing the answers to the screen or a file.

C: If faulty rules have previously inserted facts into the `DefaultTriples` or `Quads` tuple tables that mention non-existent resources, running the table scanning query will result in an error message like `Resource ID 324 cannot be resolved`. In this case, it is necessary to
transcribe the server to safely recover from the bug. Otherwise, any incorrect reasoning results can be safely eradicated without the need to transcribe (see note D).

D: To trigger rematerialization of a data store, run `remat` in the shell, or an equivalent API call. Similarly, to compact a data store, run `compact` in the shell, or use an equivalent API call.

E: If upgrading from a version prior to RDFox v7.1, see the appropriate migration guides. For example, if upgrading from v7.0, see the v7.1 migration guide. If upgrading from v6.3, see both the v7.0 and the v7.1 migration guides.

3. Requesting additional support

If any of the steps in the above flow chart results in an error that is not described, or if you have any questions about the dictionary compaction bug or the other changes in v7.1a, please contact support at Oxford Semantic Technologies via your usual channel.